Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

SEI Monographs on the Use of Commercial Software in Government Systems

Assembling Large Systems from COTS Components: Opportunities, Cautions, and Complexities

David Carney

June 1997

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **JUN 1997**	2. REPORT TYPE	3. DATES COVERED **00-00-1997 to 00-00-1997**

4. TITLE AND SUBTITLE **Assembling Large Systems from COTS Components: Opportunities, Cautions, and Complexities**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University,Software Engineering Institute,Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**14**	

About this Series

Government policies on the acquisition of software-intensive systems have recently undergone a significant shift in emphasis toward the use of existing commercial products. Some Requests for Proposals (RFPs) now include a mandate concerning the amount of COTS (commercial off-the-shelf) products that must be included. This interest in COTS products is based on a number of factors, not least of which is the spiraling cost of software. Given the current state of shrinking budgets and growing need, it is obvious that appropriate use of commercially available products is one of the remedies that might enable the government to acquire needed capabilities in a cost-effective manner. In systems where the use of existing commercial components is both possible and feasible, it is no longer acceptable for the government to specify, build, and maintain a large array of comparable proprietary products.

However, like any solution to any problem, there are drawbacks and benefits: significant tradeoffs exist when embracing a commercial basis for the government's software systems. Thus, the policies that favor COTS use must be implemented with an understanding of the complex set of impacts that stem from use of commercial products. Those implementing COTS products must also recognize the associated issues—system distribution, interface standards, legacy system reengineering, and so forth—with which a COTS-based approach must be integrated and balanced.

In response to this need, a set of monographs is being prepared that addresses the use of COTS software in government systems. Each monograph will focus on a particular topic, for example: the types of systems that will most benefit from a COTS approach; guidelines about the hard tradeoffs made when incorporating COTS products into systems; recommended processes and procedures for integrating multiple commercial products; upgrade strategies for multiple vendors' systems; recommendations about when not to use a commercial approach. Since these issues have an impact on a broad community in DoD and other government agencies, and range from high-level policy questions to detailed technical questions, we have chosen this modular approach; an individual monograph can be brief and focused, yet still provide sufficient detail to be valuable.

About this Monograph

There is a wide spectrum of systems to which a commercial off-the-shelf (COTS) approach might apply, and even the concept of what a "COTS-based system"[1] is is subject to widely varying opinions. So at the outset we must establish our frame of reference.

One end of the spectrum of COTS-based systems includes such "turnkey" systems as Microsoft Office, Common Desktop Environment (CDE), or Netscape Communicator. The capabilities that they provide are valuable, they are reasonably reliable, and they tend to fall into domains where government needs fully conform with the needs of the private sector. Systems of this type are widely and successfully used throughout many government agencies.

[1] In spite of common usage, the term "COTS" is an adjective, and we shall generally use it as such. This may entail some lengthier sentences (e.g., we will refer to "COTS products" or "a COTS-based approach"), but the greater precision of meaning is worth the cost.

Further along the spectrum, there are many other systems, particularly in the information management (IM) domain, that rely on a commercial product such as Oracle, but also have a number of customized elements specific to the given application. The commercial products generally dominate such systems, but the amount of customization can vary widely. In some cases, the customization needed by the system is straightforward and fits well with the COTS product's design and the commercial strategy of its vendor. In other cases, where the customization does not fit well or overshadows the commercial products, the system can suffer simultaneously from the constraints of a proprietary, custom-built system and the competing constraints that stem from the whims of the marketplace.

Finally, there are many systems wherein a complex mixture of commercial and non-commercial components are juxtaposed to provide large-scale functionality that is otherwise not available. These systems often have a very large amount of integrating "glue" code that binds heterogeneous pieces together, sometimes cleanly and sometimes crudely. They are often found in embedded, real-time, or safety-critical applications, and can be subsystems themselves, as, for instance, in large complex weapons or avionics systems.

For the "turnkey" systems on one end of this spectrum, an acquisition bias toward commercial products needs little justification. The growing use of such software systems in many business domains is a clear vindication of the wisdom of a commercial strategy for many government acquisitions. As we move along the spectrum, however, the decision process to use COTS—and the implementation process of building the COTS-based systems—gets progressively more complex. At the far end of the spectrum, the use of COTS products raises a large number of difficult questions.

As an example, a missile guidance control system is one whose functionality is not typically available as a commercial whole product. If it were being built a decade ago, it would probably have been written entirely "from scratch." Today, however, it may well be that some parts of that same missile system could be acquired commercially (e.g., an embedded real-time operating system, gyroscopic control software). Therefore, given the current shift toward commercial products, it might be argued these components should be purchased rather than built. But to what extent does the presence of commercial components in a missile change the way it is designed, built, and tested? Realizing that no software, commercial or otherwise, is perfect, do we entrust lives, to software for which we have neither specified its requirements nor defined its testing? And yet, given the relative costs between buying and building, can we afford not to buy? Therein lies the quandary, and it is specifically with systems of this type that this monograph is concerned.

There are other application domains, neither life-threatening nor safety-critical, but for which the issues are no less important. The Global Transportation Network, the Joint Engineering Data Management Information & Control System (JEDMICS), and the Navy Ship Design Tools program are all complex government systems for which the commercial marketplace might be a source for some of their parts. These programs inspired this paper's title, since they deal with assembling large and complex systems, often with pieces that do not fit together very well.

For program managers and technical personnel responsible for these systems who also must respond to various directives, memos, and policy statements, the simple question often arises: "OK; but what does a COTS approach mean for me? How do I 'do COTS'?" While the question may be simple, the answer is not. COTS products can bring benefits, but they can also bring a challenging set of problems and pitfalls for the people who implement the systems that use them. These problems and pitfalls stem from two sources: an unexpected degree of complexity that may result from a COTS approach, and the lack of widespread experience in dealing with that complexity.

Therefore, this monograph, the first in a series, illuminates some general issues that can arise when pursuing a COTS-based approach in complex, heterogeneous systems. Note that we do not pretend to provide an immediate and painless resolution to those issues.[2] Nor are all of these issues universally applicable; to reiterate the opening point, many systems, especially in the MIS domain, are relatively free from the problems outlined herein.

Subsequent monographs in this series will deal in greater detail with the issues mentioned in this paper. Our objective is to offer practical guidance, advice, and suggestions about these topics. But the necessary first step is to understand the intricacies that accompany the current shift in government policies toward using COTS products; that is the goal of this monograph.

Note: This first monograph in the series provides an overview of the questions and issues that arise in using COTS when assembling large complex systems of heterogeneous components. Other monographs will be published over the next several months. There will be a generally common "look-and-feel" to the series, but there will also be several distinguishing aspects. Some papers will be technical, some will be aimed at managers, and others will simply seek to build up awareness about a given question.

[2] We further assert that anyone who *does* offer simple answers is denying reality.

Assembling Large Systems from COTS Components: Opportunities, Cautions, and Complexities

1 “How do I do COTS?”

The answer to this question involves at least three things:

- deciding *whether* to use COTS products
- learning *how* to use COTS products
- gauging the *effect* of using COTS products

Note that these are not sequential steps, nor are they truly separate. In other words, making an initial decision to use COTS products in a given system may be modified as one learns about the impact on the acquisition process. This iteration will be continuous, since awareness of the effect of using COTS products will mature as systems are acquired and maintained over the course of several years, and this awareness will modify subsequent decisions for or against using COTS products.

Nonetheless, the distinction provides a useful mechanism for examining the key issues that accompany a COTS approach. We examine each in turn, principally by posing some interesting, and sometimes difficult, questions. We first consider the decision about whether (or when) to use COTS products, although this decision might well be the final one. The second and third issues, learning how to use COTS products and gauging the effect of that use, are by far the more complex actions.

2 Deciding Whether to Use COTS Products

The decision to use COTS products should not be “a given.” From a purely pragmatic perspective, this decision should be based on knowledge about the benefits and drawbacks that a COTS-based approach brings. Awareness about these benefits and drawbacks can only come from gaining experience in using COTS products; this is the principal topic that is expanded in Sections 3 and 4 below.

However, aside from any specific benefits or drawbacks, the decision to use COTS products may also result from various government policy directives. Making a decision from this point of view demands that a program manager first clarify what the various directives are actually saying, and then determine whether they are applicable to a specific acquisition or project. It also means that the manager must understand the factors that may modify decisions about using COTS products. Therefore, the decision in favor of using COTS products stems from answering the following questions: What do the directives really say? Are they applicable to my situation? What factors might modify my choices?

2.1 What Do the Directives Really Say?

A DoD program manager might well ask: What is the relevant guidance that must be followed in my situation? What are my specific obligations for using commercial products in a given system? These questions are not naive: a brief and informal survey of many DoD managers indicates that there is a varying spectrum of understanding about what is and what is not covered by official policy on COTS. Worse, there are multiple policies and directives, and the overlaps among them are not entirely free of contradiction.

Thus, a key first step is to fully understand the various directives that have been issued. In a subsequent monograph, we will examine each of these directives in detail, and will summarize their key elements, their particular sphere of relevance, and indicate any items from different directives that are in conflict.

2.2 Are the Directives Applicable?

Presuming that the policies are fully understood, the program manager must then determine which ones apply to a particular situation. Where any degree of choice exists, there must be some criteria that would indicate one choice or the other. These criteria may stem from many sources; in the future, one major source will be the growing body of experience gained in building COTS-based systems. For the present, however, the array of possible determinants is large and diverse. For instance, expectation about frequent technology refreshment for a particular system may strongly favor a COTS-based approach. Conversely, security considerations may impose stringent requirements that make a COTS-based approach impossible. Some types of systems are highly amenable to commercial solution, e.g., financial systems. Other types of systems fall into a “Defense-only” technology domain, e.g., hydroacoustic signal analysis for submarines. And the extent to which the system involves unprecedented technology may affect the decision in either direction.

Finally, a decision of this sort is seldom absolute or all-embracing. It is not likely that a complex system will be entirely composed of commercial components, so the decision is more likely to focus on the extent of commercial components in the system rather than whether to use COTS at all.

2.3 What Factors Will Modify my Choice?

While there are many factors that could modify the choice for or against using COTS products in any given system, these tend to be predictions rather than certainties. As noted at the outset, there is little experience in government (or elsewhere, for that matter) in building and maintaining large-scale COTS-based systems. However, it is prudent to speculate about the factors that will probably emerge as drivers for choosing or rejecting a COTS-based approach. As more systems are built with a significant number of COTS products, and as more individuals—managers and technologists alike—become conversant with the constraints that a COTS-based approach imposes on system development, system maintenance, and lifetime system cost, this will form the foundation of a growing experience base about the use of COTS products. The understanding gained will then contribute to better-informed decisions on whether and how to use COTS products in subsequent system development.

We discuss these predictions—which will be modified as these monographs mature—under the broad headings of learning how to use COTS and of gauging the effect of that usage.

3 Learning How To Use COTS Products

Extensive use of commercial products in government systems will potentially affect all of the traditional life-cycle stages. The familiar activities of requirements specification, design, code, integrate, test, deploy, and maintain are affected in many subtle ways when a system has a large percentage of its functionality provided by COTS products. We discuss some of the implications of these changes in Section 3.1 below.

However, even the notion of what "life cycle" means will be affected. Some activities (e.g., encapsulating, "wrapping") that have no analog in the traditional life-cycle description take on primary importance in building COTS-based systems. Other activities are analogous to traditional life-cycle steps, but are significantly different because of the use of commercial products. In Section 3.2 we will briefly examine some of these.

Note that while we can sometimes distinguish traditional from novel life-cycle stages, the distinction is not always a clean one. Thus, for instance, we can discuss traditional system integration (in Section 3.1), or we can discuss the concept of adapting and assembling COTS components (in Section 3.2). But these are clearly very similar to each other, and perhaps are really only distinct in their perspectives. In either case, it is this ambiguity that brings about unexpected difficulties in system development and maintenance.

3.1 Some Traditional Life-Cycle Activities

Whether it is the "waterfall," the "spiral," or any other traditional life-cycle model, there is a set of fairly well-understood activities that occur when acquiring systems. The inclusion of a focus on COTS products brings some pervasive changes to these activities. We will briefly examine some of the impacts that a COTS approach has on requirements, testing, and maintenance.

The familiar method of defining requirements is essentially straightforward: One describes a desired system through a set of specified conditions that the system must meet. However, defining requirements is very different when acquiring COTS-based systems, since at least some software requirements must be flexible enough to accommodate the fluctuations of the commercial marketplace. In such cases, either the requirements will be written to describe existing products, or so that they are malleable enough to be implemented with a variety of existing products. For instance, if a system involving some CASE tools includes a project management tool, and the expectation is for a COTS item from most bidders, then the author of its requirements must have adequate knowledge of the existing CASE marketplace, which will then guide the description of required functional features. Anything else would be self-contradictory (e.g., soliciting bids for a commercial product, yet describing functional capabilities for which no commercial instances exist).

Testing is also a different activity in a COTS-based approach. Since a COTS component is essentially a black box, it may be very difficult to determine what types of testing, either at the unit level or at the system level, are possible or necessary. Yet a system designer will be faced with the reality of using a COTS component: How should one determine what testing will be necessary? The requirements that drove the vendor's creation of that component may or may not be documented; if the latter is true, it may be very difficult to gauge whether they are appropriate for the system at hand.

Finally, maintenance changes in a COTS-based system. Upgrading a COTS-based software system means that as new releases of the commercial components are made by the various vendors, the system may incorporate them. A system with several commercial components thus has a very heavy dependence on various release cycles of the COTS vendors. A further complicating factor is that different pieces of the system will be upgraded at widely varying intervals; licenses will need to be revalidated for different parts of the system at random intervals. And multiple component upgrades can result in numerous unforeseen problems—incompatible files and databases, different naming conventions, introduction of new conflicts between COTS components—these problems are not at all uncommon. Depending on the number of COTS components and different COTS vendors, the effect of these multiple dependencies can vary from short-term user inconvenience to total system instability.

3.2 Some Less Familiar Life-Cycle Activities

Some key differences between a traditional life cycle and a COTS-based one can be seen in Figure 1, which shows the types of activities that apply to COTS-based systems. Some of these activities are new, having no counterpart in the traditional life cycle. Others are similar to those discussed in the previous section, yet are different enough that they are almost unprecedented activities.

In the life cycle depicted by this model, the commercial marketplace supplies an assortment of products, from which the system builder selects some components to examine more closely. These must be evaluated to determine which are fit for use in the system to be built. Since experience has shown that commercial products will rarely fit together without some sort of third-party work, it is likely that the system builder will next adapt the chosen products, which can be expensive, time-consuming, or both. (Note that adaptation does not imply modification, but the effort involved can range from being trivial to huge.) Then the adapted components are assembled, a step that involves a complex interaction of architecture, infrastructure, and possibly middleware elements. Finally, since the commercial marketplace is continually in flux, updates to the commercial components will need to be made (possibly often).

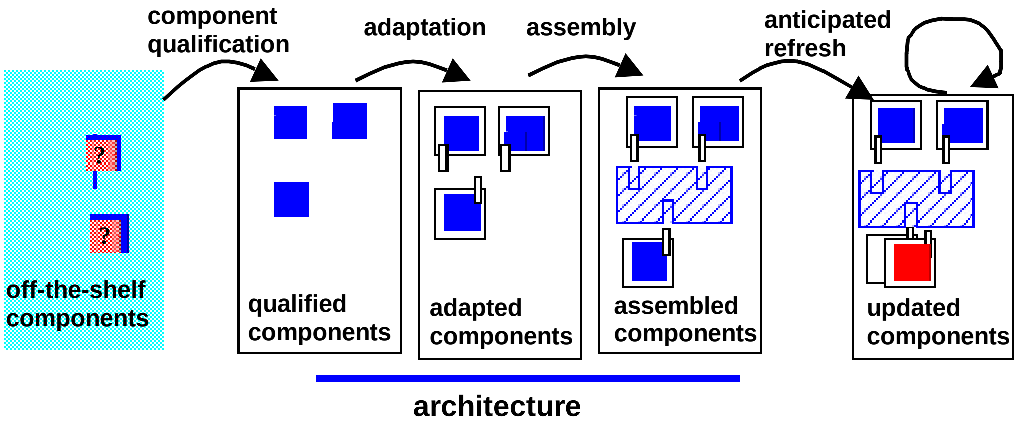

Figure 1: Creating COTS-Based Systems

Each of these activities places some unfamiliar demands on both managers and technologists. For instance, selecting and qualifying appropriate COTS products means that the commercial marketplace, currently large and rapidly growing larger, must be constantly surveyed in anticipation of new system development. This would appear to be an ongoing expense with little apparent justification. Yet to do otherwise (i.e., to perform a reasonable market survey “from scratch” under the pressure of an acquisition’s specific schedule) is probably imprudent; it will certainly be difficult.

For performing product evaluations, guidelines and methods are needed. But software evaluation is an area in which there is little wisdom on which to rely. When viable candidate products have been identified, someone must do tradeoff analyses between competing products, a task that is not necessarily straightforward. The notion of evaluation also raises questions at a more fundamental level. What are the decision criteria for migrating to new or emerging technologies at all? When choosing between two implementations of a relational database, a more basic question should first be answered: What is the basis for the use of relational technology at all? Should an object-oriented database be used instead? And if so, why?

Adapting and assembling COTS products also presents unique difficulties. The realities of the software world are quite different from those of the hardware world, and COTS software components are seldom built to “plug” into each other easily. The usual way to overcome this deficiency and build integrated systems out of incompatible COTS software components involves “wrappers,” “bridges,” or other “glueware.” However, this technique does not necessarily lead to lower costs. Writing wrappers can be a complex activity, requiring expertise both at the detailed system level and in the COTS components being wrapped. The net result is that this can increase, rather than decrease, a system’s overall cost

Underlying the entire COTS-based life cycle, architecture also has a role in developing a system. But this role may not be the same as in a traditional life cycle. Choosing an architecture as the basis for a system will probably be a subtly different exercise when the components and their interfaces are outside of one’s control. There is a converse issue as well: How is a system’s architecture the result of the available choices? How do the twin concepts of architecture and integration interplay? These questions demand that we create guidelines for building systems by composing rather than constructing. And given the assertion that a only part of a system will likely be composed of commercial components, this also implies that we need guidelines for evaluating a commercial product’s potential for integration with a custom legacy system. It also implies the need to understand the risks and risk mitigation strategies for doing so.

Finally, system maintenance and evolution has long been known to be the most expensive portion of the life-cycle cost, and we have already discussed (in Section 3.1) how maintenance activities are deeply affected by extensive use of COTS components. Yet it is in this area of evolution (or “anticipated refresh” in the diagram above) that so little experience exists. Can we determine the quality of long-term system support that a given vendor will provide? Can we predict the longevity of the commercial vendor? What is the fallback position when the vendor of a critical component goes out of business?

These questions are neither new nor profound; to a large extent, they have been relevant in some acquisitions for years (e.g., questions like these are always in the mind of a buyer of CASE tools). But as systems become more and more tied to the commercial marketplace, the relevance of such questions grows, as does the need for good answers to them.

4 Gauging the Effect of Using COTS Products

Of the many effects that might follow from extensive use of COTS products, three are of particular interest here. First, regardless of the amount of COTS products that it contains, the system itself still requires engineering; this does not come for free. Second, the use of COTS products has a profound effect on the entire process of building a business case and costing an acquisition. Third, one of the most pervasive (and largely unexpected) results of extensive use of COTS is that it demands a paradigm shift in many quarters, not merely technical. We briefly examine each of these below.

4.1 The Ongoing Need to Engineer the System Well

The discipline of engineering is no less critical to a COTS-based system than any other type of system; in some circumstances it could be even more critical. The reality of today’s available COTS products is that few of them are designed to work together. Many have been created to be stand-alone products, and to require no co-location (let alone interaction) with any other product or component. Even when they have been designed to cooperate with another product, it is most often another product from the same vendor or from another vendor with whom the first vendor shares some special interest.

A COTS-based system is still a system with its own requirements, both developmental and life-cycle. Although the parts might be obtained from commercial sources, no one cares about the system itself except the people who will pay for it, maintain it, and use it. This system must be designed, brought together, tested, and managed just the same as any other system that was built or acquired in the past. There are no magic formulae for this. Nor is the government’s responsibility for its systems eliminated by the new-found reliance on COTS products.

4.2 The Business Case for Using COTS Products

Although the motivation for widespread use of COTS products is cost savings, there are many unknowns that must be addressed from a business perspective. For instance, we have briefly referred to the unforeseen expense needed to keep appropriate “wrappings” on COTS products throughout the entire maintenance phase. How should a program manager react if a commercial approach results in a higher life-cycle cost? What business case should be made in that circumstance?

There are equally difficult questions that focus on the commercial marketplace itself. For instance, what are the decision factors that affect the choice of vendor? How does a program manager guess the potential cost (and risk) of letting a critical component, its maintenance, and its upgrade cycle be under the control of the commercial marketplace? What are the measurements by which we can compare the ongoing upgrade costs of various products from different vendors?

Finally, the move toward COTS products is paralleled by the move toward open systems; but how are these two affected by each other? A COTS-based system and an open system are not the same thing (although the policy shift toward commercial components stems from the same impulse as that toward standards), and there may be points where they will be in conflict. Suppose, for instance, that in evaluating two products, one is technically superior but based on a proprietary standard, the other technically inferior but based on an open standard. Which is the better choice? What are the short- and long-term costs associated with either choice?

4.3 The Unforeseen Need to Bring About a Large-Scale Paradigm Shift

In most college curricula today, the introduction to software and computer science still consists of learning one or more programming languages. This teaches people to write whole systems and subsystems, designing them from a blank piece of paper, then coding, debugging, and testing them.

In contrast, the use of existing products as components in a system requires determination of how to get them to cooperate with one another to achieve the goals of the system. This will often result in writing wrappers to achieve the desired cooperation and integration. And this will almost certainly eventuate in repeating this step during maintenance as each product changes (usually independently) to keep the set of components continuously cooperating.

These two paradigms are very different, and the move to the generation of COTS-based systems constitutes a significant paradigm shift for programmers and system developers. Extrapolating from that, we find that it also constitutes a significant paradigm shift for the testing, quality assurance, and maintenance personnel as well. And changes to all these positions require changes and paradigm shifts in managers, in the expectations they have, and in the techniques they employ.

In other words, the change to COTS-based systems is not just a technological change. It affects many people in many roles in profound ways. Organizations can be equally affected, experiencing changes in the activities they undertake, their structure and their relationships, required training, the corporate policies, the relationships between government and contractors, and relationships across the marketplace. This paradigm shift toward integration of others' products, from a producer to a consumer mentality, has widespread effects. The worst thing one can do is to treat it as merely as a change in technology.

5 Summary: "But *How* Do I Do COTS"?

At the beginning of this paper, we promised to

- make some observations
- indicate some difficulties
- provide some high-level suggestions

The first two of these promises have been relatively easy to keep. However, the third promise has not yet been fulfilled, and you are probably still asking: "But *how* do I do COTS?" We now provide two suggestions in reply.

The first is no more than another promise, namely, that over the next several months, we plan to address many of the issues raised in this paper. Our hope and expectation is to provide pragmatic advice and recommendations, and to deal with the many problems of developing COTS-based systems.

However, as a second answer to the “How do I do COTS” question, there are numerous practical things that can be done in the short term. While stated at a still very general level, we suggest that the following are actions that any project manager might immediately undertake. They do not form a “roadmap,” but they do act as a set of five practical first thoughts that will prove useful when starting the journey toward COTS-based systems.

1. Start now to educate yourself about regulations and obligations.

Know the regulations as they emerge in as great a detail as you can. If possible, keep a “legal guru” on hand whose ongoing work is to be aware of the various government regulations.

2. Start now to educate yourself about some relevant subset of the commercial marketplace.

The size of the COTS product marketplace is huge and growing, at least in some domains. Since no one can be conversant in the entire marketplace, it is reasonable to assign one or more people to become conversant in a useful and interesting subset of it. To some extent, this represents a gamble. But if your next project somehow falls into a different domain, you are no worse off than if you did nothing, and you may well have some valuable perspectives on how the commercial marketplace functions.

3. Use previous projects as proving grounds for trying on a “COTS perspective.”

Past projects (those that are at least beyond initial deployment) can be valuable “dry runs.” For instance: Examine the requirements specification and imagine that a mandate to use COTS products as much as possible would have been in force. Which requirements would this mandate affect? How would a product survey be done? What products are known to be available? Given the schedule of the acquisition, how much time would be available for surveying the marketplace?

4. Rethink any existing system maintenance activities in light of a “COTS perspective.”

Existing systems that are being maintained, updated, or revised can be an equally useful proving ground. Assume, for instance, that some avionics system under your control has an ongoing series of bug fixes, and that your current approach involves a team of programmers that fixes them and periodically makes new releases. Now assume that the system is a COTS system, and that bug fixes are under the vendor’s control. How does this affect the operations of the fielded system? What impact might this have on scheduling any updates of other systems in the aircraft?

5. Start now to develop a database of metrics: costs, time spent in all activities, personnel records, and anything else that might provide a later basis for comparison of a COTS-based approach with a traditional one.

One of the most fundamental aspects of the policy shift toward COTS products is the expectation of cost savings. At the moment, there is little actual data to verify this, and such data is vitally needed. The data is needed about both custom systems and COTS-based ones, since it is in the comparison of the two that a COTS approach will be either validated or not.

Afterword

This introductory monograph is intended to define the boundaries of discourse for the rest of the papers in the series by addressing at a high level the different areas in which questions arise and for which answers must be found. Other monographs will be published over the next several months. There will be a generally common "look-and-feel" to the series, but there will also be several distinguishing aspects. Some papers will be technical, some will be aimed at managers, and others will simply seek to build up awareness about a given question. In some cases, multiple papers on a single topic will target both a managerial and well as various technical points.

While these monographs will not provide all-embracing answers to all questions, each will provide some insight into a given topic by raising hard issues that must be faced. In some cases, the answers must then be found by the people "in the trenches" whose responsibility it is to implement government systems.

The full set of topics is still under consideration. While the final choices are yet to be made, it is likely that they will include topics similar to the following:[3]

- finding and selecting appropriate commercial products
- assessing the flexibility and malleability of system requirements
- guidelines and methods for performing product evaluations
- evaluating the potential for integrating commercial products with an existing system
- decision criteria for migrating to new or emerging technologies
- variance between traditional testing approaches and those needed for COTS-based systems
- the role of architecture in COTS-based systems
- developing a commercial outlook on system maintenance

We expect that the list of titles will evolve and change; it is also likely that a given monograph may be revised and reissued as more experience is gained in a given area.

[3] Note that this is not a prioritized list, and makes no commitment about content or order of publication

Feedback

Comments or suggestions about these monographs are welcome. We want this series to be responsive to the real needs of government personnel. To that end, comments concerning inclusion of other topics, the focus of the papers, or any other issues are of great value in continuing this series of monographs. Comments should be sent to:

Editor
SEI Monographs on COTS
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
cots@sei.cmu.edu